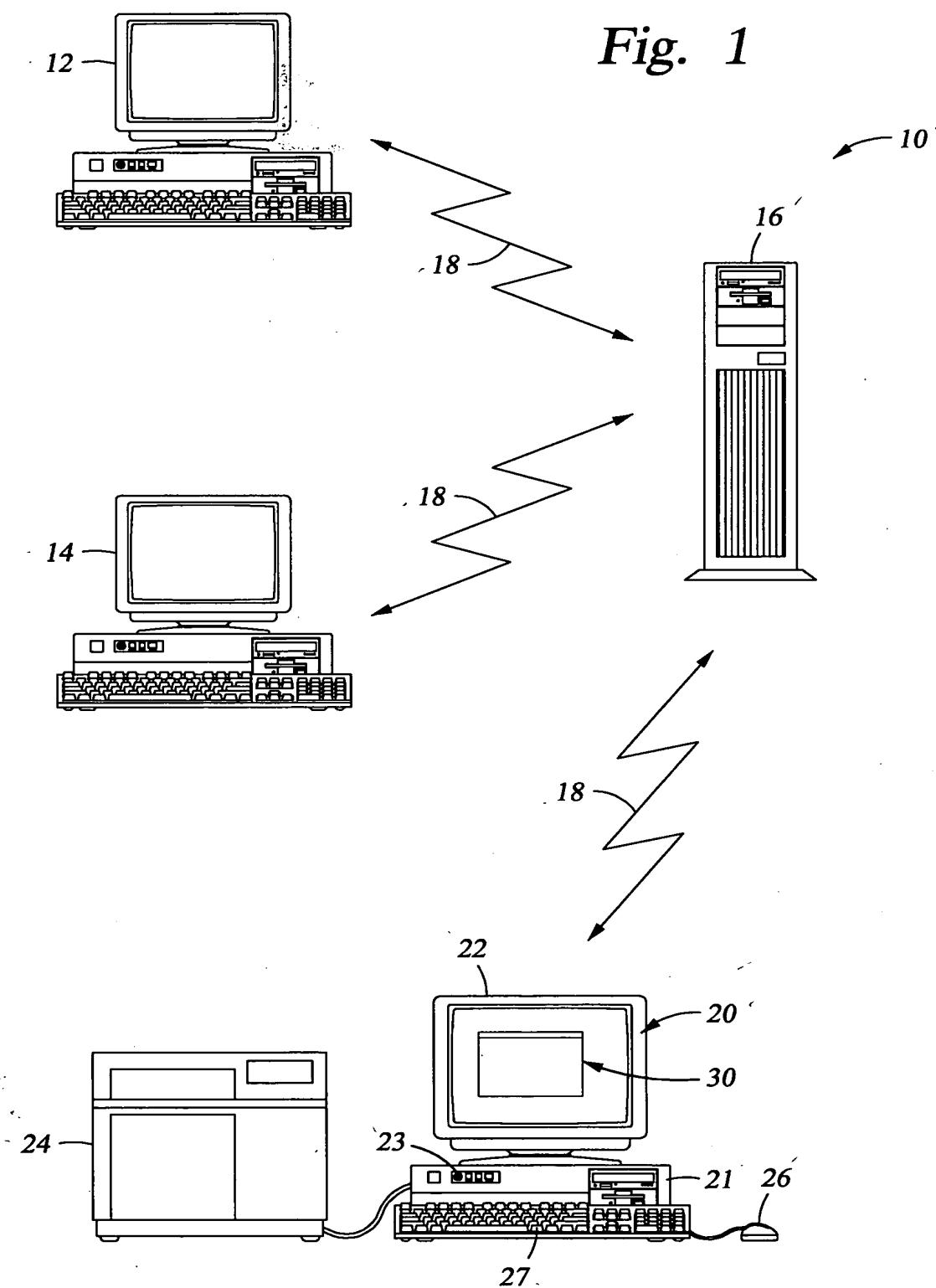


Fig. 1



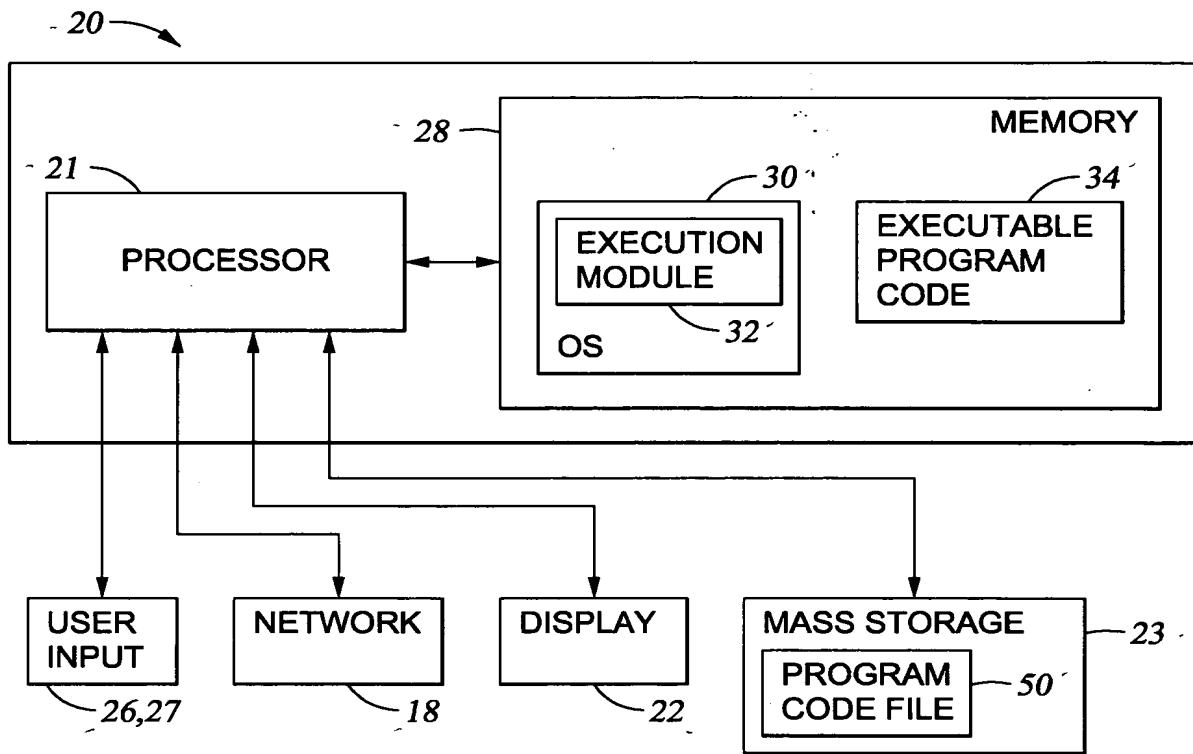


Fig. 2

Direct Call

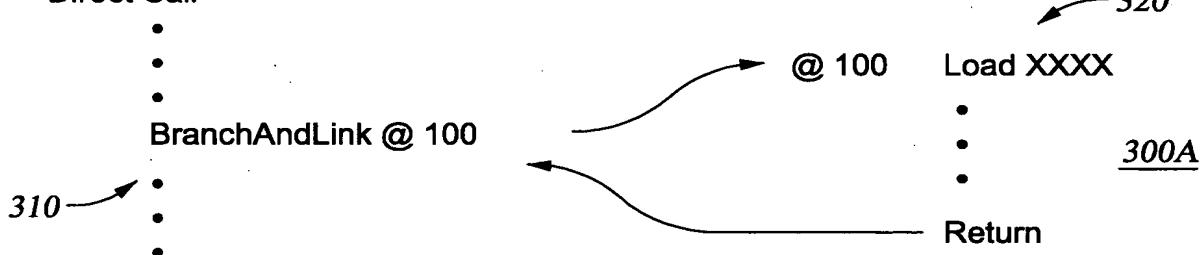


Fig. 3A

Indirect Call

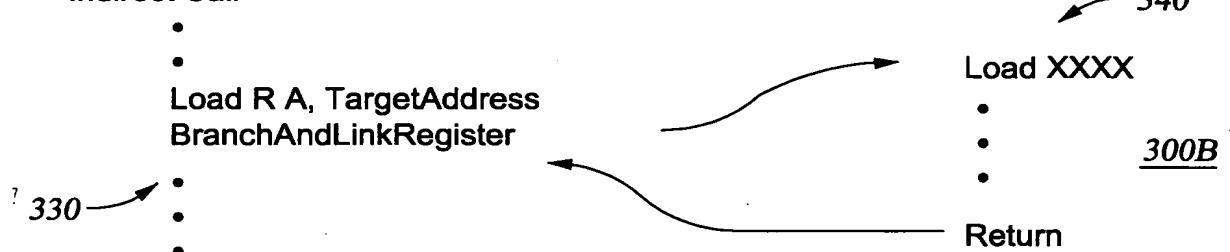


Fig. 3B

3/15

Inlining

•
•
•
Load R1,X
Load R2, Y
BranchAndLink @ 100 → @100 Add R3,R1,R2
•
•
•
Return

•
•
•
Load R1,X
Load R2, Y
Add R3,R1,R2

•
•
•

Fig. 4

Inlining (Continued)

•
•
•
LoadImmediate R1,5
LoadImmediate R2,6
BranchAndLink @ 100 → @100 Add R3,R1,R2
•
•
•
Return

•
•
•
LoadImmediate R3,11
•
•
•

Fig. 5

4/15

MULTIPLE COPIES OF STATIC

Fig. 6A

```
Class A {  
    static int X= 5;  
    static synchronized int getNext() { 600A  
        X++;  
        return X;  
    }  
}
```

Fig. 6B

```
class Aprime {  
    static int X= 5;  
    static synchronized int getNext() { 600B  
        X++;  
        return X;  
    }  
}
```

Virtual Call

-
-
-

Load RA, offset of MethodRef in constant pool(RS)
Load RB, offset of VMT offset in MethodRef(RA)

Load RG, Ø (RO)
Add RD, RB, RC
Load RE, Ø (RD)

1310

Load RF, offset to entry point ptr in MethodRef(RE)
Move RP0, RE
Move RT,RF
BranchIndirect

Fig. 13

SEPARATELY COMPILED COPIES OF THE SAME CLASS

```
class A {  
    static int doSomthing() {  
        int Y = B.someMethod() - C.anotherMethod();  
        •  
        •  
        •  
    }  
}
```

Static as compiled for first copy:

-
-
-

@ 11 ptr to B.someMethod
@ 12 ptr to C.anotherMethod

-
-
-

700

Static as compiled for fsecond copy:

-
-
-

@ 11 ptr to C.anotherMethod
@ 12 ptr to B.someMethod

-
-
-

Fig. 7

TRADITIONAL COMMON ADDRESS SPACE SCHEME

```
class A {
    static int X;
    static int doSomething() {
        X = B.someMethod() - C.anotherMethod();
        •
        •
        •
    }
}
```

	code	Static
@ 123	Call @ 11	@ 10 X
	Call @ 12	@ 11 ptr to B.someMethod
	Add	@ 12 ptr to C.anotherMethod
	Store @ 10	

810 → ← 820 → 830

Fig. 8A

TRADITIONAL COMMON REENTRANT IMAGE SCHEME

(Same source)

Code	Static
Call @ 11 (RS)	RS →
Call @ 12 (RS)	•
Add	•
Store 10(RS)	•
	10
	11
	12

840 → ← 850

Fig. 8B

900 CONTENTS OF A JAVA CLASS FILE

- 910 -	CLASS DESCRIPTION
- 920 -	STATIC FIELD DEFINITIONS
- 930 -	INSTANCE FIELD DEFINITIONS
- 940 -	METHOD DEFINITIONS
- 950 -	CONSTANT POOL

Fig. 9

1000 STRUCTURE OF A LOADED JAVA CLASS

- 910 -	CLASS DESCRIPTION
- 920 -	STATIC FIELD DEFINITIONS
- 930 -	INSTANCE FIELD DEFINITIONS
- 940 -	METHOD DEFINITIONS
- 950 -	CONSTANT POOL
- 1060 -	STATIC STORAGE
- 1070 -	VIRTUAL METHOD TABLE

Fig. 10

1100 STRUCTURE OF A COMPILED JAVA CLASS

- 910 -	CLASS DESCRIPTION
- 920 -	STATIC FIELD DEFINITIONS
- 930 -	INSTANCE FIELD DEFINITIONS
- 940 -	METHOD DEFINITIONS
- 950 -	CONSTANT POOL
- 1110 -	VERIFICATION DIRECTIVES
- 1120 -	CONSTANT RESOLUTION ~ DIRECTIVES
- 1130 -	CODE GENERATION STATIC
- 1140 -	CODE

Fig. 11

Indirect Call

-
-
-
- Load RA, offset of MethodRef in constant pool(RS)
- Load RB, offset of MethodInfo ptr in MethodRef(RA)
- Load RC, offset to entry point ptr in MethodInfo(RB)
- Move RP0 RB
- Move RT, RC

BranchAndLink Indirect

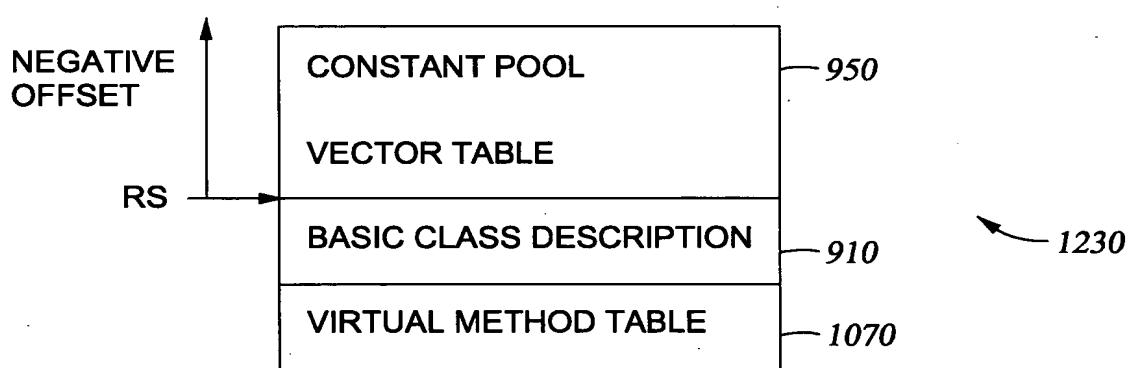
-
-
-

1210 ↗

Direct Call

-
-
-
- Load RA, offset of MethodRef in constant pool(RS)
- Load rb, offset of MethodInfo ptr in MethodRef(RA)
- Move RP0, RB
- BranchAndLink Direct @ of entry

1220 ↗

*Fig. 12*

© 1995 Microsoft Corporation. All rights reserved.

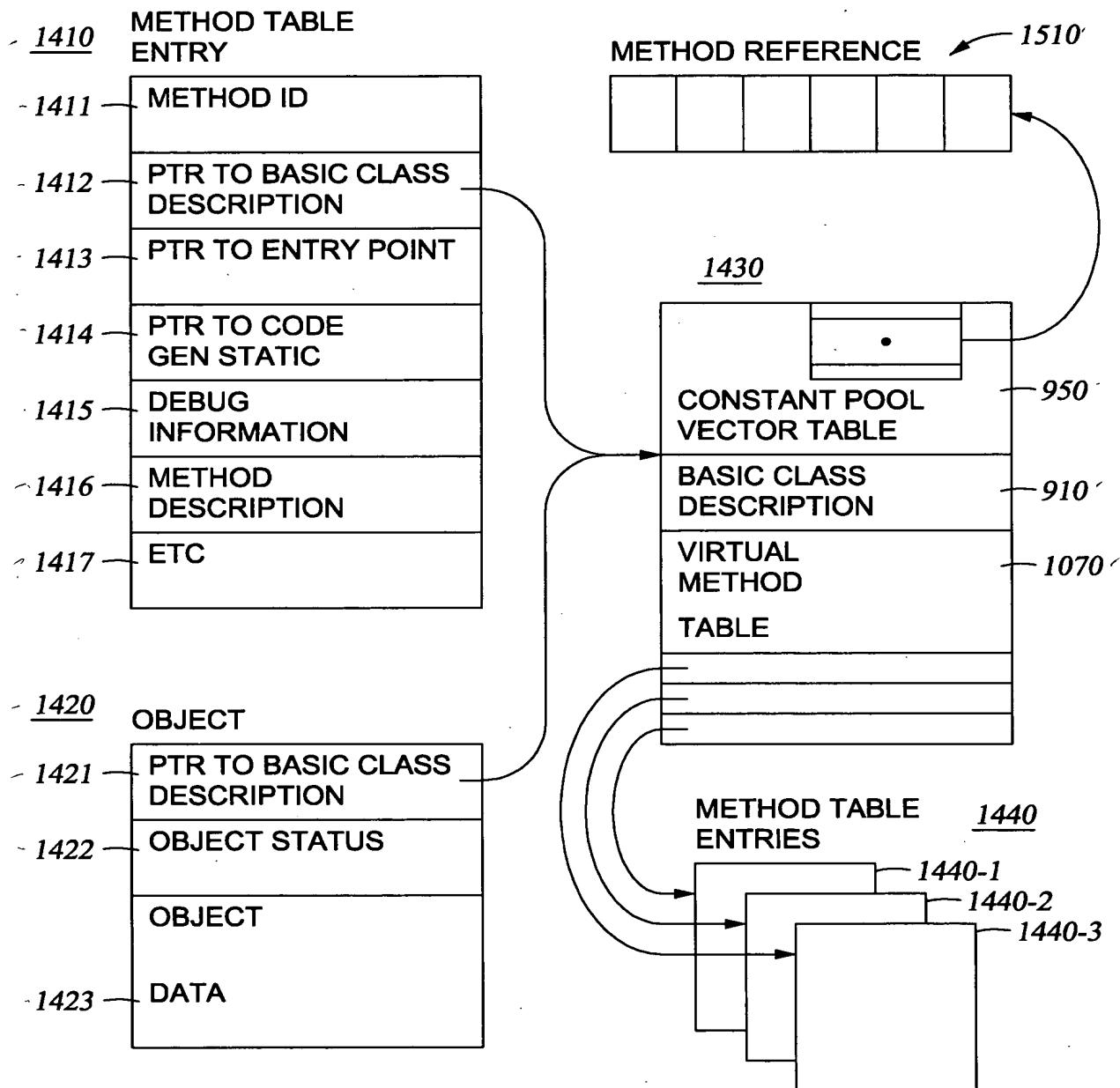


Fig. 14

10/15

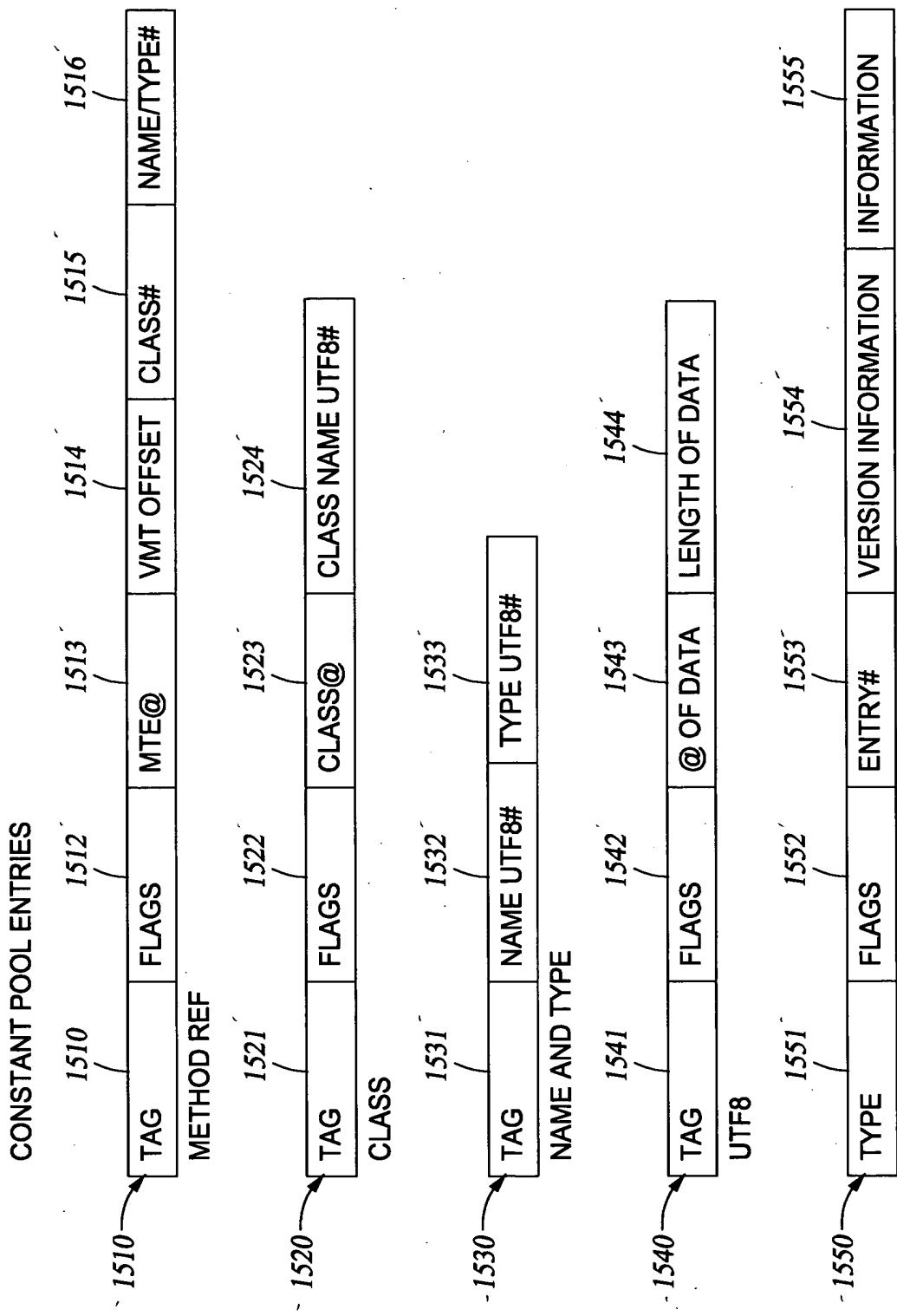


Fig. 15

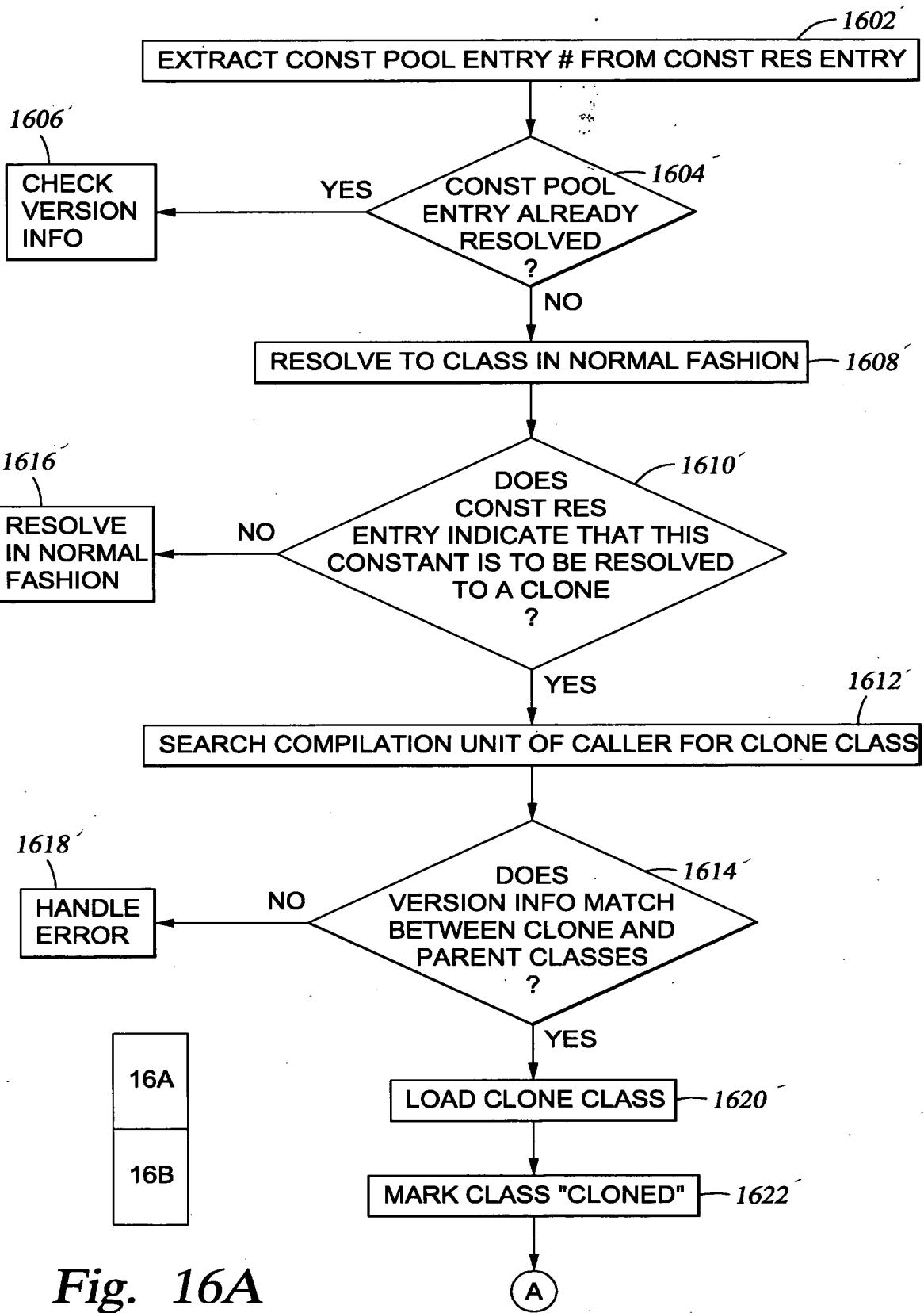


Fig. 16A

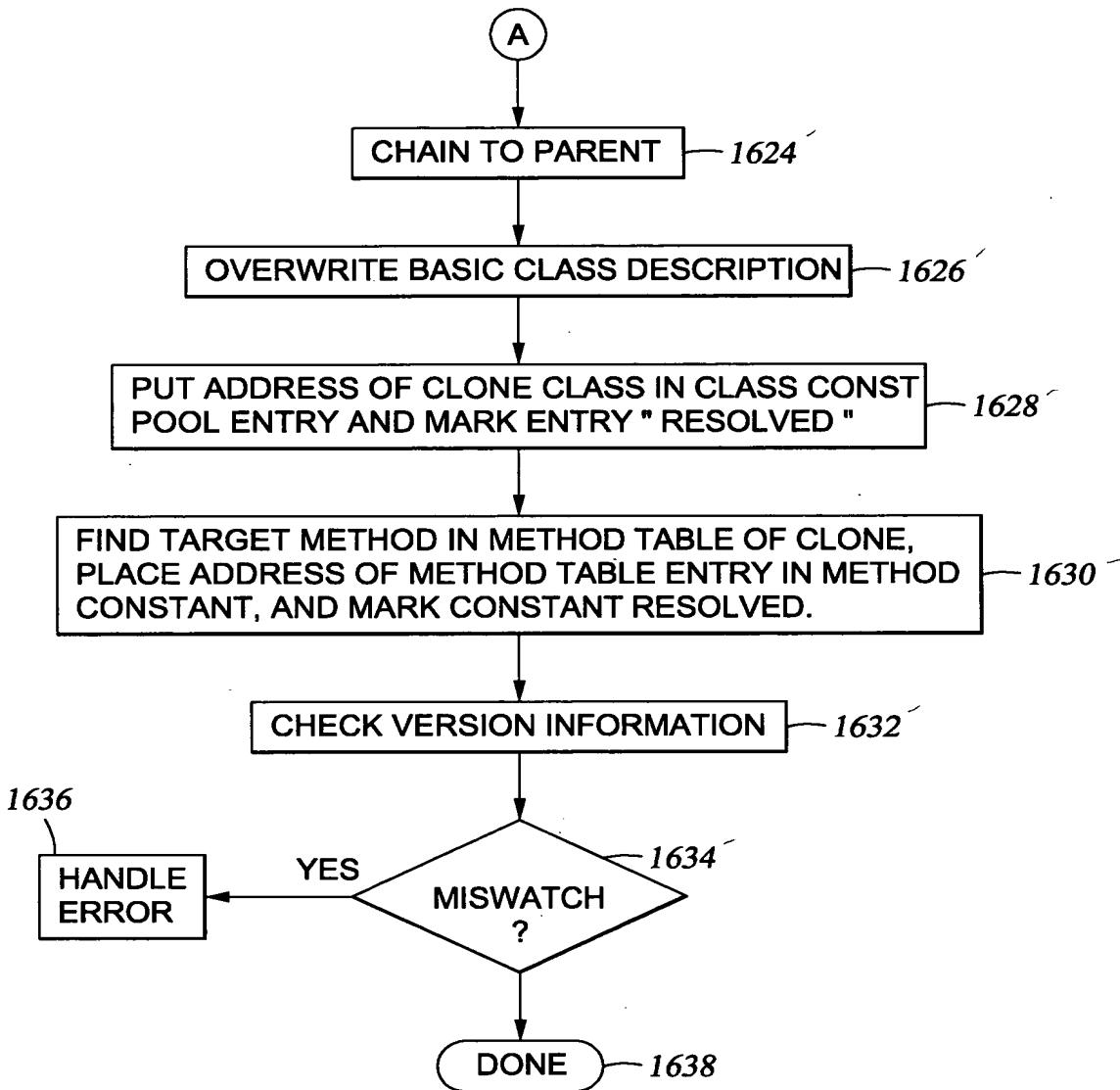


Fig. 16B

13/15

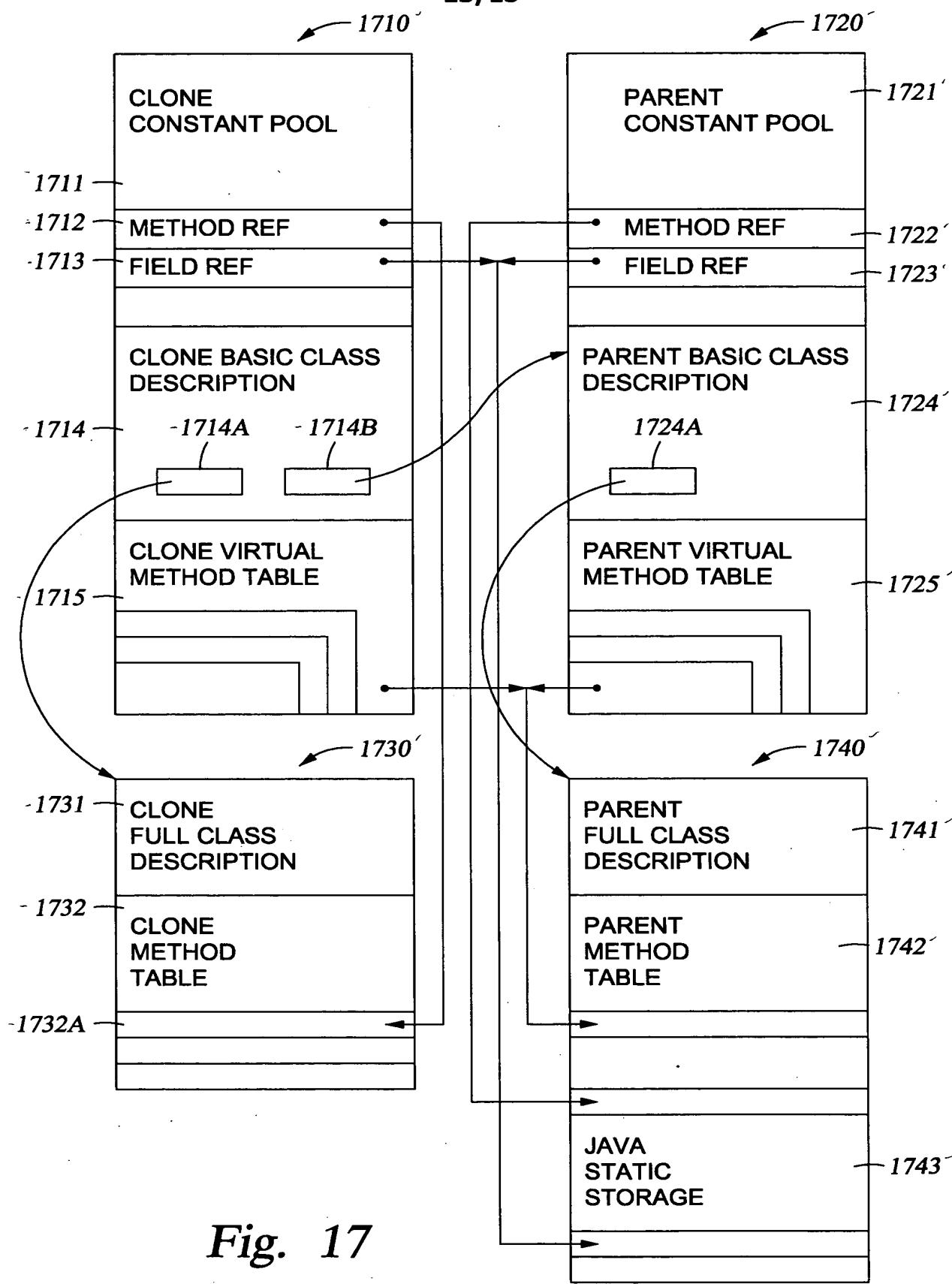


Fig. 17

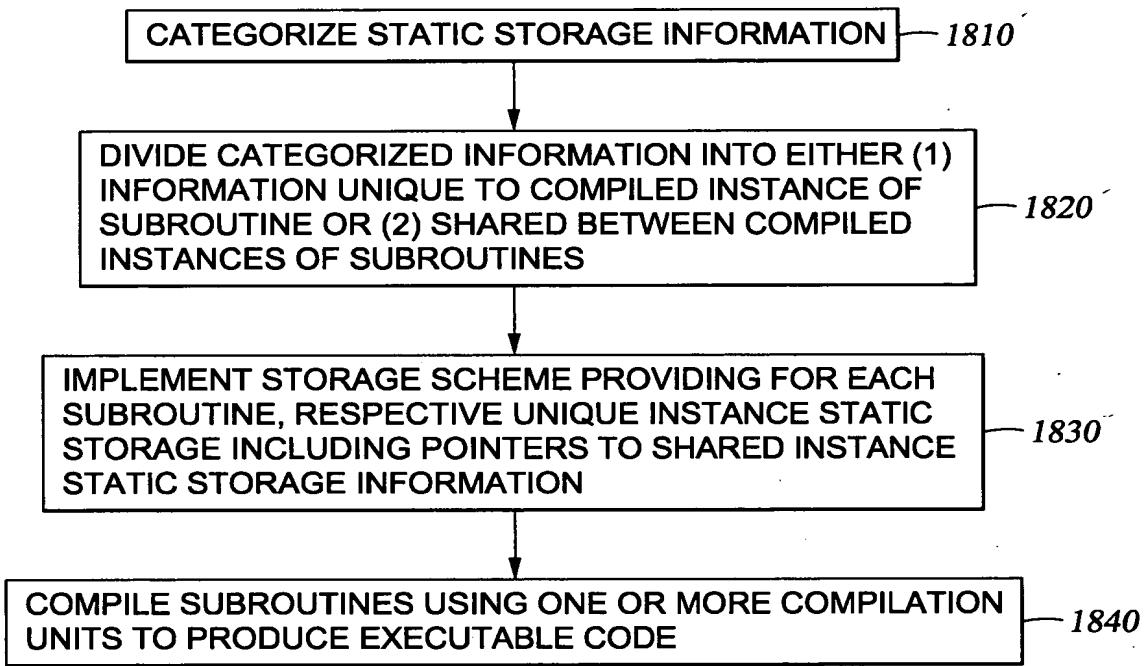


Fig. 18

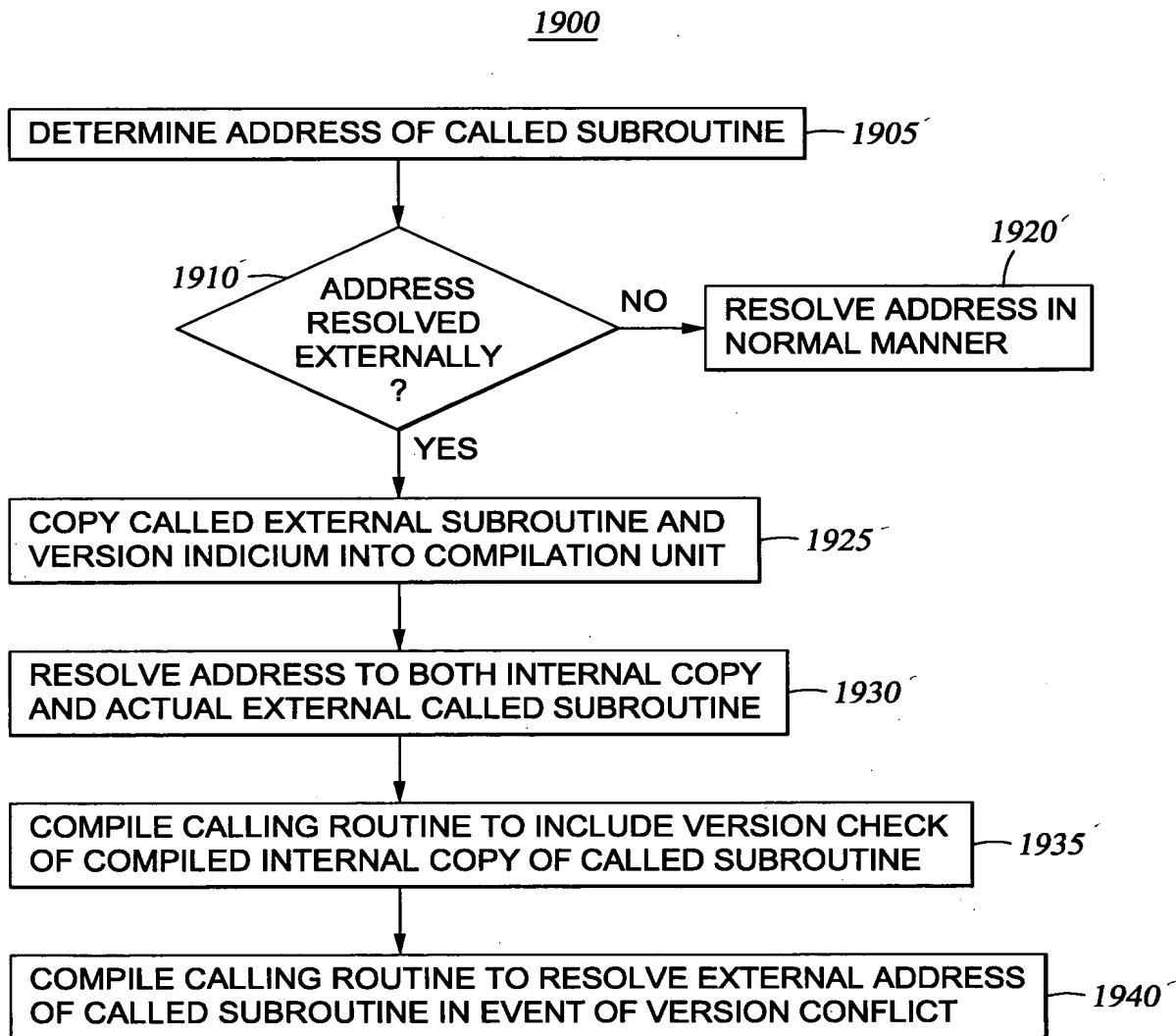


Fig. 19